

REMARKS

Claims 1-18 are pending. Claims 1 and 13 have been amended.

As requested by the Examiner, the specification has been amended to update the reference to the related application referenced on page 7, line 2.

Rejection Under 35 U.S.C. § 112

Claims 1, 13 and 14 stand rejected under 35 U.S.C. § 112, first paragraph, as failing to comply with the written description requirement. The Examiner asserts that the claimed phrase “wherein more than one of said plurality of versions of a said software application *executing on said server and* available to service requests from users on said server” is not described in the original filed disclosure. In support of this assertion, the Examiner points to the description of spawning a server process on pages 7 and 12 of the specification.

However, as is described in more detail beginning at the bottom of page 11 and carrying over to page 12, the process that the server spawns is a communications process that will pass the request and response between the requester and the version of the application running on the server:

In most preferred embodiments the server will *spawn a communications process to handle the REQ and the RESP to be returned* 75. The communications process or other program within the server handles connecting the REQ to the *SWVx program [i.e., the requested version of the application]* in accord with the information in the table. The *SWVx then process the REQ and sends out the information required for a RESP* (response) either through the communications process or other mechanism available in the server 78.

Specification, p. 12 (emphasis added). Thus, the reference to “spawning a process” isn’t a reference to the application program.

With respect to the application program, the Summary of the Invention makes it clear that the applications are executing on the server:

A plurality of versions of software application programs can be handled by a single server serving multiple user-clients who each need access to specific ones of the plurality of versions. Thus such different versions *can run simultaneously* without requiring upgrading of early versions and no interference between versions.

Specification, pp. 2-3 (emphasis added). Thus, the applicants respectfully submit that claims 1, 13 and 14 do satisfy the written description requirement. Reconsideration of the Section 112 rejection is respectfully requested.

Rejection Under 35 U.S.C. § 101

Claims 1-13 stand rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter. The applicants have amended claims 1 and 13 to overcome the rejection. Reconsideration is respectfully requested.

Rejection Under 35 U.S.C. § 103

Claims 1, 3-11 and 13-18 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Ciarlante et al. (6,532,488) (hereinafter “Ciarlante”) in view of Nishiyama et al. (5,859,977) (hereinafter “Nishiyama”). Claims 2 and 12 stand rejected under Section 103(a) as being unpatentable over Ciarlante and Nishiyama and further in view of Bryan et al. (6,591,418) and Mutschler et al. (5,974,430), respectively. Reconsideration is respectfully requested.

The present invention allows a server computer to run different versions of a software application program in such a way that clients of the server can issue requests for service and have those requests serviced by a selected one of the versions of the application program running on the server. *See* Summary of the Invention, pp. 2-3 (“[a] plurality of versions of software application programs can be handled by a single server serving multiple user-clients who each need access to specific ones of the plurality of versions”). The Examiner incorrectly asserts that Ciarlante and Nishiyama together teach all of the features of claims 1, 13 and 14.

Ciarlante describes a system that offers to host different applications from different software vendors to groups of users. The system essentially enables a company or other entity to “rent” a software application without having to purchase its own copy of the software and install it on the company’s servers. Instead, the system of Ciarlante will host the software application on its own server for the company.

As shown in Fig. 13, for example, the system will present a variety of different application program “offerings” to a user (*i.e.*, a company), such as an “Auction” application,

a “Real Estate” application that manages real estate listings, or a “Planning/Purchasing” application. When the company selects a particular application program offering, an instance of that application is created on the system’s server and users can then use the hosted application from client computers via the Internet. The Examiner attempts to draw similarities between the system of Ciarlante and the claimed invention, but the systems are really very different.

Most importantly, as admitted by the Examiner, “Ciarlante fails to disclose plurality versions of a software application and accessing by SiteID, and table containing SiteID” (Office Action, ¶ 8). However, the Examiner asserts that “Nishiyama shows a system maintenance and management of different version of the same application, and also provide details of site management functions with table containing SiteID (fig 3 and fig 14, col 7, line 42 to col 8, line 52)” and that “[i]t would have been obvious ... to incorporate the detailed teachings of site management of Nishiyama into the software application hosting system taught by Ciarlante to provide access to different copies of software application over a wide area network.” The problem with this argument is that the claimed invention is not a “site management system” like that described in Nishiyama, and no combination of Nishiyama with Ciarlante would produce the claimed invention. Nishiyama has nothing to do with hosting multiple, different versions of a software application on a single server, as claimed. Rather, Nishiyama describes a system that manages the distribution of software to various *physical* sites (*i.e.*, computers) in a network.

Specifically, as described in Nishiyama, a “software distribution management table” is used to keep track of which version of a software application has been distributed to a given physical site and also the particular method used to distribute the software to that site (*e.g.*, bulletin board, broadcast or download). The purpose of the system is to manage software upgrades throughout the network. Specifically, as explained in Nishiyama,

The software distribution management table 301 ... is a table for *managing software and its version and revision distributed to respective sites*.

In a site name field 302, site names managed by the software distribution management table 301 are stored. In a software name field 303, names of all softwares in the system managed by the software distribution management table 301 are stored as "software-1, software-2, . . . ". This software name field 303 is further divided into two parts, *i.e.*, a management kind field 304 and a version number field 305. The management kind of the above described

software and the identification information of matter to be distributed are stored in the management kind field 304. *As for the management kind, the above described bulletin board method, broad cast method, and down load method are denoted by 1, 2 and 3, respectively.* As for the matter to be distributed, source and program (having execute form) are denoted by S and P, respectively. . . .

At the time of distribution, the version number of the software which has been distributed is stored in the version number field 305 of the pertinent software, i.e., the software-1 in case of FIG. 3, in the software distribution management table 301.

By using the method heretofore described, version/revision management of software of the information system becomes possible.

Col. 7, ln. 44 – col. 8, ln. 21 (emphasis added). As further explained by Nishiyama,

In the site management function of the present embodiment, program attributes such as name of software stored in each site, version and revision, occupation size in the memory, and resident (which means that the program is stored on the main memory) or nonresident (which means the program is read from an external memory such as a disk every time the program is started) are also managed. *In the network computer system, the site management function makes it possible to immediately discriminate the version and revision of each site and thereby prevent a mistake at the time of program replacement.*

Col. 8, lines 36-52 (emphasis added). As these portions of Nishiyama demonstrate, Nishiyama's software version/revision management system has nothing to do with the applicants' claimed system and method for servicing of user requests by different versions of a software application program running on a server.

Nowhere does Nishiyama teach or suggest receiving a user request that includes a SiteID and passing that request to a particular one of a plurality of versions of a software application program executing on a server in order to have that version of the software service the request. On the contrary, Nishiyama is concerned with managing software upgrades in a network, not servicing user requests issued to a server. Because of the significant differences between Nishiyama and the claimed invention, no combination of Ciarlante and Nishiyama would produce the claimed invention. Nor do Bryan or Mutschler cure the deficiencies of those primary references.

For the foregoing reasons, the applicants respectfully submit that the Examiner has failed to state a prima facie case of obviousness under 35 U.S.C. § 103(a). Reconsideration of the Section 103(a) rejection of independent claims 1, 13 and 14 is therefore respectfully

DOCKET NO.: USYS-0200/RA5373
Application No.: 09/738,852
Office Action Dated: March 12, 2008

**PATENT
REPLY FILED UNDER EXPEDITED
PROCEDURE PURSUANT TO
37 CFR § 1.116**

requested. Inasmuch as claims 2-12 and 15-18 each depend either directly or indirectly from one of the independent claims, the applicants submit that they too patentably define over the art of record for the same reasons.

CONCLUSION

For all the foregoing reasons, the applicants respectfully submit that the present application is now in condition for allowance.

Respectfully submitted,

Date: July 14, 2008

/Steven B. Samuels/
Steven B. Samuels
Registration No. 37,711

Woodcock Washburn LLP
Cira Centre
2929 Arch Street, 12th Floor
Philadelphia, PA 19104-2891
Telephone: (215) 568-3100
Facsimile: (215) 568-3439

Please address all correspondence to:
Richard Gregson, Esq.
Patent Counsel
Unisys Corporation
Unisys Way
Blue Bell, PA 19424
Telephone: (215) 986-4411
Facsimile: (215) 986-3090